

REMARKS

Claims 1-57 are pending in the present application. Claims 1, 6-9, 14, 16, 20, 25-28, 33, 35, 39, 44-47, 52 and 54 are amended for clarity. Support for the amendments to the claims may be located at least on page 10, lines 20-27 and on page 14, line 14 through page 17, line 3. Reconsideration of the claims is respectfully requested.

I. 35 U.S.C. § 112, Second Paragraph

The Office Action rejects claims 1-57 under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter, which applicants regard as the invention. This rejection is respectfully traversed.

The Office Action states:

As to claims 1-57, they are not clearly indicated one of object in objects for one protocol or one object for two or more different protocols.

Amendments are made to claims 1, 6-9, 14, 20, 25-28, 33, 39, 44-47, and 52 to clarify that a request for object data is received according to a protocol of a protocol interface. The protocol is one of the two or more different protocols that the OID abstraction layer is able to interpret. The OID abstraction layer interprets the request according to the protocol from the protocol interface, identifies/locates a repository where the object data is maintained and converts the request to a format for the API that communicates with the repository. Once the OID abstraction layer receives the object data retrieved from the repository through the API, the object data is encapsulated in a reply message and the reply message is formatted for an appropriate protocol for the protocol interface. The appropriate protocol is one of the two or more different protocols. Therefore, Applicant respectfully submits that the rejection of claims 1-57 under 35 U.S.C. § 112, second paragraph is overcome.

II. 35 U.S.C. § 103, Alleged Obviousness Based on *Spofford* and *Dobbins*

The Office Action rejects claims 1, 20 and 39 under 35 U.S.C. § 103(a) as being allegedly unpatentable over *Spofford et al.* (U.S. Patent 5,913,037), hereinafter referred to as *Spofford*, in view of *Dobbins et al.* (U.S. Patent 5,951,649), hereinafter referred to as *Dobbins*. This rejection is respectfully traversed.

As to independent claims 1, 20 and 39, the Office Action states:

As to claim 1, Spofford teaches the invention substantially as claimed including: OID (OID, col 2, ln 59-67, col 6, ln 1-45, col 4, ln 1-9, col 7, ln 20-62, col 8, ln 15-52), abstraction layer (MIB manager, col 2, ln 59-67/ col 6, ln 1-45/col 4, ln 1-9/ col 7, ln 20-62/ col 8, ln 15-52/ col 11, ln 1-30/ col 12, ln 40-67), an OID tree structure (col 2, ln 59-67/ col 6, ln 1-45, col 4, ln 1-9/ col 7, ln 20-62/ col 8, ln 15-52/ col 11, ln 1-30/ col 12, ln 40-67), query (query, col 11, ln 1-15), repository (the MIB 206, col 9, ln 40-41/ col 10, ln 58-59).

Spofford does not explicitly teach the OID abstraction layer is capable of receiving queries for objects in two or more different protocols, registering the ODI tree structure with a registry associated with the OID. However, Dobbins teaches the OID abstraction layer is capable of receiving queries for objects in two or more different protocols (a standard interface for the Management Information Base for object access by any management protocol or other entity including SNMP, SNMPv2, DMP ..., col 16, ln 20-23), registering the ODI tree structure with a registry associated with the OID (Each specific managed object which is persistent is then created and calls the Persistent Object Manager 77 to store the value, col 20, ln 33-39/ all BaseResources are registered into one of these tables for management purposes, col 24, ln 49-53).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine teaching of Spofford and Dobbins because Dobbins's the OID abstraction layer is capable of receiving queries for objects in two or more different protocols, registering the ODI tree structure with a registry associated with the OID would provide a high availability of service, remoter management for supporting a number of different routing protocols.

As to claims 20, 39, they are apparatus claims of claim 1; therefore, they are rejected for the same reason as claim 1 above.

Office Action dated December 14, 2004, pages 3-4.

Claim 1, which is representative of the other rejected independent claims 20 and 39 with regard to similarly recited subject matter, reads as follows:

1. A method on a server in a distributed data processing system for maintaining a logical composite repository of Object Identifier (OID) tree structures, the method comprising the steps of:

receiving, in an OID abstraction layer, an OID tree structure from a repository, wherein the OID abstraction layer is capable of receiving queries for objects in two or more different protocols and supports the two or more different protocols by mapping queries from multiple protocol interfaces to application program interface (API) requests that the repository understands;

registering the OID tree structure with a registry associated with the OID abstraction layer; and

adding the OID tree structure to a repository associated with the OID abstraction layer. (emphasis added)

Neither *Spofford* nor *Dobbins*, either alone or in combination, teaches or suggests an OID abstraction layer that is capable of receiving queries for objects in two or more different protocols and supports the two or more different protocols by mapping queries from multiple protocol interfaces to application program interface (API) requests that the repository understands.

Spofford is directed to a dynamic management information base manager. A management information base (MIB) manager allows agents to add or delete objects to any level within the MIB tree by object identifier (OID). The MIB manager is a set of software interfaces, semantics, procedures, and data structures that work together to dynamically manage a tree of simple network management protocol (SNMP) data objects identified by an OID along with each object's value. SNMP is the only management protocol contemplated by *Spofford*. As stated in the Office Action, *Spofford* does not teach or suggest an OID abstraction layer that is capable of receiving queries for objects in two or more different protocols. Further, *Spofford* does not teach or suggest an OID abstraction layer that is capable of receiving queries for objects in two or more different protocols and supports the two or more different protocols by mapping queries from multiple protocol interfaces to application program interface (API) requests that the repository understands, as recited in claims 1, 20 and 39.

Dobbins is directed to a network interconnecting apparatus having a separate forwarding engine object at each interface. Each forwarding engine only knows the configuration information and how to receive and transmit packets on the one interface to which it corresponds. Each forwarding engine acts independently to process packets, yet each interacts together to collectively provide packet forwarding. An object-oriented architecture is provided that distributes the critical function and system behavior into self-contained router objects. All router objects have the functions provided by the managed object class, which defines the methods and data for network management, built into the router object. The services and data normally external to the object are embedded or accessible within the object itself. *Dobbins* does not teach or suggest an OID abstraction layer that is capable of receiving queries for objects in two or more different protocols and supports the two or more different protocols by mapping queries from

multiple protocol interfaces to application program interface (API) requests that the repository understands, as recited in claims 1, 20 and 39.

In the rejection of independent claims 1, 20 and 39, the Office Action refers to the following portion of *Dobbins*:

A standard interface for the Management Information Base for object access by any management protocol or other entity including SNMP, SNMPv2, DMP, local device management, and other Managed Objects.

Dobbins, column 16, lines 20-23.

Although this portion of *Dobbins* states that any management protocol may be used to access an object using a standard interface for the Management Information Base (MIB), further analysis of *Dobbins* shows that *Dobbins* does not teach that multiple management protocols may be used to access an object using the standard interface for the MIB. Specifically, Figure 3A and Figure 3B of *Dobbins* reference an SNMP agent 228. Additionally, column 29, lines 31-33 of *Dobbins* states "SNMP operates by passing request to a device's internal database, the Management Information Base (MIB)." Thus, the preferred embodiment of *Dobbins* uses SNMP as the management protocol, but another management protocol may be used instead of SNMP. *Dobbins* does not teach or suggest an OID abstraction layer that is capable of receiving queries for objects in two or more different protocols. Further, *Dobbins* does not teach or suggest an OID abstraction layer supports the two or more different protocols by mapping queries from multiple protocol interfaces to application program interface (API) requests that the repository understands, as recited in claims 1, 20 and 39.

Additionally, *Dobbins* teaches that services and data normally external to the object are embedded or accessible within the object itself. Thus, there would not be a need for an OID abstraction layer to map queries from multiple protocol interfaces to application program interface (API) requests that the repository understands. Further, as stated in the Office Action, *Spofford* and *Dobbins* do not teach API.

Neither *Spofford* nor *Dobbins* teach or suggest an OID abstraction layer that is capable of receiving queries for objects in two or more different protocols and supports the two or more different protocols by mapping queries from multiple protocol interfaces to application program interface (API) requests that the repository understands.

Therefore, the alleged combination of *Spofford* and *Dobbins* does not teach or suggest this feature, as recited in independent claims 1, 20 and 39.

Thus, neither *Spofford* nor *Dobbins*, either taken alone or in combination, teaches or suggests an OID abstraction layer that is capable of receiving queries for objects in two or more different protocols and supports the two or more different protocols by mapping queries from multiple protocol interfaces to application program interface (API) requests that the repository understands, as recited in independent claims 1, 20 and 39. Accordingly, Applicant respectfully requests withdrawal of the rejection of claims 1, 20 and 39 under 35 U.S.C. § 103(a).

III. 35 U.S.C. § 103, Alleged Obviousness Based on *Spofford*, *Dobbins* and *Whitehead*

The Office Action rejects claims 2-4, 21-23 and 40-42 under 35 U.S.C. § 103(a) as being allegedly unpatentable over *Spofford* in view of *Dobbins* and further in view of *Whitehead* et al. (U.S. Patent 6,085,030), hereinafter referred to as *Whitehead*. This rejection is respectfully traversed.

Since claims 2-4, 21-23 and 40-42 depend from independent claims 1, 20 and 39, respectively, the same distinctions between *Spofford* and *Dobbins*, and the invention recited in claims 1, 20 and 39, apply to dependent claims 2-4, 21-23 and 40-42. In addition, *Whitehead* does not provide for the deficiencies of *Spofford* and *Dobbins* with regard to independent claims 1, 20 and 39. *Whitehead* is directed toward a network component server that provides an object-neutral global component registry. *Whitehead* is cited for teaching an anchor point. *Whitehead* does not teach or suggest an OID abstraction layer that is capable of receiving queries for objects in two or more different protocols and supports the two or more different protocols by mapping queries from multiple protocol interfaces to application program interface (API) requests that the repository understands. Thus, any alleged combination of *Whitehead* with *Spofford* and *Dobbins* still would not result in the invention recited in claims 1, 20 and 39 from which claims 2-4, 21-23 and 40-42 depend. Accordingly, Applicant respectfully requests withdrawal of the rejection of claims 2-4, 21-23 and 40-42 under 35 U.S.C. § 103(a).

IV. 35 U.S.C. § 103, Alleged Obviousness Based on *Spofford*, *Dobbins* and *Ferguson*

The Office Action rejects claims 5-8, 9-18, 24-27, 28-37, 43-46 and 47-56 under 35 U.S.C. § 103(a) as being allegedly unpatentable over *Spofford* in view of *Dobbins* and further in view of *Ferguson* (U.S. Patent 6,016,499). This rejection is respectfully traversed.

As to independent claims 9, 28 and 47, the Office Action states:

As to claim 9, *Spofford* teaches a first query (a query, col 10, ln 25-67 to col 11, ln 1-16), the object data (the objects, col 10, ln 25-67 to col 11, ln 1-16), a request (request, col 10, ln 25-67), a protocol (SNMP, col 1, ln 1-35/ protocol, col 5, ln 5-67/ col 6, ln 1-67), OID (OID, col 2, ln 59-67, col 6, ln 1-45, col 4, ln 1-9, col 7, ln 20-62, col 8, ln 15-52), abstraction layer (MIB manager, col 2, ln 59-67/ col 6, ln 1-45/ col 4, ln 1-9/ col 7, ln 20-62/ col 8, ln 15-52/ col 11, ln 1-30/ col 12, ln 40-67).

Spofford does not explicitly teach the OID abstraction layer is capable of receiving queries for objects in two or more different protocols, locating a repository that contain the object data requested in the first query based on a registry. However, *Dobbins* teaches the OID abstraction layer is capable of receiving queries for objects in two or more different protocols (a standard interface for the Management Information Base for object access by any management protocol or other entity including SNMP, SNMPv2, DMP, col 16, ln 20-23), locating a repository that contain the object with that database, by using the object's identifier (OID) ... the format of these requests by providing a textual representation to these OID's, which are easier for the user to digest, col 29, ln 34-40 and col 42-45).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine teaching of *Spofford* and *Dobbins* because *Dobbins*'s the OID abstraction layer is capable of receiving queries for objects in two or more different protocols, registering the ODI tree structure with a registry associated with the OID would provide a high availability of service, remoter management for supporting a number of different routing protocols. ...

As to claims 24-28, 29-37, 43-47, 48-56, they are apparatus claims of claims 5-9, 10-18; therefore, they are rejected for the same reason as claims 5-9, 10-18 above.

Office Action dated December 14, 2004, pages 5-8.

Claim 9, which is representative of the other rejected independent claims 28 and 47 with regard to similarly recited subject matter, reads as follows:

9. A method on a server in a distributed data processing system for retrieving object data from a repository, comprising:
receiving a first query for the object data from a requester in the distributed data processing system, wherein the first query is in a protocol recognized by an OID abstraction layer; wherein the OID abstraction layer is

capable of receiving queries for objects in two or more different protocols and supports the two or more different protocols by mapping queries from multiple protocol interfaces to application program interface (API) requests that the repository understands;

interpreting the first query according to the protocol recognized by the OID abstraction layer, wherein the protocol recognized by the OID abstraction layer is one of the two or more different protocols;

locating a repository that contains the object data requested in the first query based on a registry associated with the OID abstraction layer; and
retrieving the object data from the repository using an OID abstraction layer application program interface (API). (emphasis added)

As discussed above, *Spofford* and *Dobbins*, either taken alone or in combination, fail to teach or suggest an OID abstraction layer that is capable of receiving queries for objects in two or more different protocols and supports the two or more different protocols by mapping queries from multiple protocol interfaces to application program interface (API) requests that the repository understands, as recited in independent claims 1, 9, 20, 28, 39 and 47. In addition, *Ferguson* does not provide for the deficiencies of *Spofford* and *Dobbins* with regard to independent claims 1, 9, 20, 28, 39 and 47. Further, *Spofford*, *Dobbins* and *Ferguson*, either alone or in combination, do not teach or suggest interpreting the first query according to the protocol recognized by the OID abstraction layer, wherein the protocol recognized by the OID abstraction layer is one of the two or more different protocols, as recited in claims 9, 28 and 47.

Ferguson is directed to a system and method for accessing a directory services repository. *Ferguson* does generally teach application program interfaces (API) and, more specifically, teaches an API that includes at least one callable element that is capable of accessing a component of a repository in response to being called and a driver that is capable of translating a database language statement, such as an SQL statement, into an executable API sequence. However, *Ferguson* does not teach or suggest an OID abstraction layer that is capable of receiving queries for objects in two or more different protocols and supports the two or more different protocols by mapping queries from multiple protocol interfaces to application program interface (API) requests that the repository understands, as recited in claims 1, 9, 20, 28, 39 and 47. Further, *Ferguson* does not teach or suggest interpreting the first query according to the protocol recognized by the OID abstraction layer, wherein the protocol recognized by the OID abstraction

layer is one of the two or more different protocols, as recited in independent claims 9, 28 and 47.

Spofford, Dobbins and Ferguson do not teach or suggest an OID abstraction layer that is capable of receiving queries for objects in two or more different protocols and supports the two or more different protocols by mapping queries from multiple protocol interfaces to application program interface (API) requests that the repository understands, as recited in independent claims 1, 9, 20, 28, 39 and 47. Therefore, the alleged combination of *Spofford, Dobbins and Ferguson* does not teach or suggest this feature, as recited in independent claims 1, 9, 20, 28, 39 and 47.

Additionally, *Spofford, Dobbins and Ferguson* do not teach or suggest interpreting the first query according to the protocol recognized by the OID abstraction layer, wherein the protocol recognized by the OID abstraction layer is one of the two or more different protocols. Therefore, the alleged combination of *Spofford, Dobbins and Ferguson* does not teach or suggest this feature, as recited in claims 9, 28 and 47.

Thus, *Spofford, Dobbins and Ferguson*, taken individually or in combination, do not teach or suggest an OID abstraction layer that is capable of receiving queries for objects in two or more different protocols and supports the two or more different protocols by mapping queries from multiple protocol interfaces to application program interface (API) requests that the repository understands, as recited in independent claims 1, 9, 20, 28, 39 and 47 and do not teach or suggest interpreting the first query according to the protocol recognized by the OID abstraction layer, wherein the protocol recognized by the OID abstraction layer is one of the two or more different protocols, as recited in independent claims 9, 28 and 47. *Spofford, Dobbins and Ferguson*, taken individually or in combination, do not teach or suggest the features of dependent claims 5-8, 10-18, 24-27, 29-37, 43-46 and 48-56 at least by virtue of their dependency on claims 1, 9, 20, 28, 39 and 47, respectively. Accordingly, Applicant respectfully requests withdrawal of the rejection of claims 5-8, 9-18, 24-27, 28-37, 43-46 and 47-56 under 35 U.S.C. § 103(a).

In addition, with regard to amended claims 6, 25 and 44, *Spofford, Dobbins and Ferguson*, either taken alone or in combination, do not teach or suggest that the OID abstraction layer receives the information retrieved from the repository through the API and encapsulates the information in a reply message to a target protocol interface,

wherein the reply message is formatted for an appropriate protocol for the target protocol interface, and wherein the appropriate protocol is one of the two or more different protocols. The cited portions of *Spofford* and *Ferguson* only teach a protocol interface, a request, a reply message and an API rather than teaching that the reply message is formatted for an appropriate protocol for the protocol interface, and wherein the appropriate protocol is one of the two or more different protocols, as recited in claims 6, 25 and 44.

Additionally, with regard to amended claims 7-8, 26-27 and 45-46, the cited portion of *Ferguson* only teaches translating the API result into a relational database result. *Spofford*, *Dobbins* and *Ferguson*, either taken alone or in combination, do not teach or suggest that the OID abstraction layer receives a request for object data from a requesting protocol interface, interprets the request according to a protocol of the requesting protocol interface, wherein the protocol is one of the two or more different protocols, converts the request into an application program interface (API) request that is forwarded to the repository, and receives an API reply from the repository having the object data, as recited in claims 7, 26 and 45. The applied references also fail to teach or fairly suggest that the OID abstraction layer reformats the object data in a reply message according to the protocol and sends the reply message to the protocol interface, as recited in claims 8, 27 and 46.

With regard to amended claims 14, 33 and 52, Applicant submits that *Spofford*, *Dobbins* and *Ferguson*, either taken alone or in combination, do not teach or suggest that the first reply is transformed into a second reply, wherein the second reply is consistent with the protocol for the first query recognized by the OID abstraction layer, and wherein the protocol recognized by the OID abstraction layer is one of the two or more different protocols. With regard to amended claims 16, 35 and 54, Applicant submits that *Spofford*, *Dobbins* and *Ferguson*, either taken alone or in combination, do not teach or suggest that each repository in the plurality of repositories contains information representing an Object Identifier (OID) subtree structure, and wherein the plurality of repositories are formatted to support the two or more different protocols. Thus, in addition to being dependent on their respective independent claims, claims 5-8, 9-18, 24-

27, 28-37, 43-46 and 47-56 are also distinguished over the *Spofford, Dobbins* and *Ferguson* references based on the specific features recited therein.

V. 35 U.S.C. § 103, Alleged Obviousness Based on *Spofford, Dobbins, Ferguson* and Admitted Prior Art

The Office Action rejects claims 19, 38 and 57 under 35 U.S.C. § 103(a) as being allegedly unpatentable over *Spofford* in view of *Dobbins* and further in view of *Ferguson* and further in view of Admitted Prior Art (APA). This rejection is respectfully traversed.

Since claims 19, 38 and 57 depend from independent claims 9, 28 and 47, respectively, the same distinctions between *Spofford, Dobbins, Ferguson* and the invention recited in claims 9, 28 and 47, apply to dependent claims 19, 38 and 57. Accordingly, Applicant respectfully requests withdrawal of the rejection of claims 19, 38 and 57 under 35 U.S.C. § 103(a).

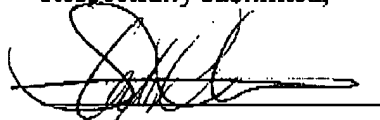
Further, there is no suggestion or motivation whatsoever in *Spofford, Dobbins, and Ferguson* for using CIM/XML. The Office Action states that *Spofford, Dobbins, and Ferguson* do not teach CIM/XML. The mere fact that a prior art reference can be readily modified does not make the modification obvious unless the prior art suggested the desirability of the modification. *In re Laskowski*, 871 F.2d 115, 10 U.S.P.Q.2d 1397 (Fed. Cir. 1989) and also see *In re Fritch*, 972 F.2d 1260, 23 U.S.P.Q.2d 1780 (Fed. Cir. 1992) and *In re Mills*, 916 F.2d 680, 16 U.S.P.Q.2d 1430 (Fed. Cir. 1993). The Office Action may not merely state that the modification would have been obvious to one of ordinary skill in the art without pointing out in the prior art a suggestion of the desirability of the proposed modification. In this case, the only suggestion or motivation for making the proposed modification is found in Applicant's own specification. As a result, absent any teaching, suggestion, or incentive from the prior art to make the proposed modification, the presently claimed invention can be reached only through the an impermissible use of hindsight with the benefit of Applicant's disclosure a model for the needed changes.

VI. Conclusion

It is respectfully urged that the subject application is patentable over the cited reference and is now in condition for allowance. The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: March 14, 2005

Respectfully submitted,



Stephen R. Tkacs
Reg. No. 46,430
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Agent for Applicant

SRT/vja